tions only include format information, and a NOP instruction is inserted whenever no instruction that cannot be executed in parallel is present. In such an arrangement, the major effect of the present invention, namely, the ability to indicate instructions using an instruction format of only the necessary length is still achieved.

(6) As can be seen from the instruction formats shown in FIGS. 6A~6F, the above embodiments describe a case where only part of a constant operand can be positioned in the second of the two units used to compose a 42-bit instruction, although an opcode may alternatively be positioned into this unit. As a result, the construction shown in FIG. 5 may be changed so that the unit that was directly outputted as part of the constant operand may be inputted into the instruction decoder, and the input bit width of the instruction decoder may be increased.

(7) In the above embodiments, the instruction buffer was described as having the construction shown in FIG. 8, although the present invention is not restricted to this construction or to this buffer size. As one example, one instruction buffer with a simple queue structure may be used.

(8) Software that achieves the functioning of the instruction conversion apparatus described in the second embodiment may be distributed having been stored on a recording medium such as a floppy disk, a hard disk, a CD-ROM, an MO (Magnetic-Optical) disc, or a DVD (Digital Versatile Disc).

The executable program generated by the instruction conversion apparatus of the above embodiments of the present invention may be distributed having been recorded onto a floppy disk, a hard disk, a CD-ROM, an MO disc, a DVD, or a semiconductor memory.

Although the present invention has been fully described by way of examples with reference to accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Therefore, unless such changes and modifications depart from the scope of the present invention, they should be construed as being included therein.

What is claimed is:

1. An instruction conversion apparatus that converts an instruction sequence into parallel execution codes that are executable by a target processor, the target processor having predetermined limitations regarding combinations of instructions capable of being executed in parallel,

the instruction conversion apparatus comprising:

assigning means for successively assigning instructions in the instruction sequence to parallel execution codes; and

control means for controlling the assigning means so that a combination of a plurality of instructions that have already been assigned to a parallel execution code and an instruction that the assigning means is about to assign to the parallel execution code satisfy the predetermined limitations of the target processor;

wherein the target processor includes (1) a fetch means for successively fetching parallel execution codes that each include a plurality of unit fields from outside the target processor, (2) s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch means, (3) decoding means, including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers, the decoders decoding at least one opcode stored in any of the 1^{st} to s^{th} registers, and (4) operation executing means, connected to the s+k−1 registers for executing operations in accordance with a decoding result of the s decoders,

the assigning means assigning, when instructions to be assigned to a parallel execution code include a long instruction whose word length is equal to at least two but no more than k unit fields, one of an opcode and an operand of the long instruction to a u^{th} (where u is any integer such that $1<u<s$) unit field between the 1^{st} unit field and the s^{th} unit field, and only an operand of the long instruction to unit fields from a $(u+1)^{th}$ unit field to a $(u+k-1)^{th}$ unit field.

2. The instruction conversion apparatus of claim 1, further comprising:

grouping means for forming an instruction group of a plurality of instructions that do not exhibit a dependency relation (hereafter "data dependency relation"), a data dependency relation being a relation between an instruction defining a resource and an instruction referring to the same resource; and

first detecting means for detecting, when a 1^{st} to an s^{th} unit field in a parallel execution code have been assigned at least one instruction by the assigning means and an instruction (hereafter "short instruction") with a shorter word length than a long instruction is left in the instruction group, a long instruction assigned to unit fields between the 1^{st} unit field and the s^{th} unit field,

wherein the control means includes a first control unit for controlling the assigning means to rearrange instructions that have already been assigned to the parallel execution code so that the detected long instruction is assigned to unit fields between the s^{th} unit field and the $(s+k-1)^{th}$ unit field and the short instruction remaining in the instruction group is assigned to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

3. The instruction conversion apparatus of claim 2,

wherein the instruction group includes instructions that exhibit an anti-dependence and instructions that exhibit an output dependence, an anti-dependence being a relation between an instruction that refers to a resource and an instruction that thereafter defines the resource, and an output dependence being a relation between an instruction that defines a resource and another instruction that defines the resource,

the control means including a search unit for searching for a combination pattern, composed of a plurality of instructions in the instruction group, that is unaffected by an anti-dependence and an output dependence, and

the first control unit controlling the assigning means to rearrange the plurality of instructions in accordance with the combination pattern found by the search unit, to assign the long instruction found by the detecting means to unit fields from the s^{th} unit field to the $(s+k-1)^{th}$ unit field, and to assign a short instruction left in the instruction group to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

4. The instruction conversion apparatus of claim 3, further comprising:

flag setting means for setting a parallel execution boundary flag at each boundary that marks a position at which the predetermined limitations of the target processor dictate that parallel execution is not possible.

5. The instruction conversion apparatus of claim 4, further comprising:

address resolving means for assigning a real address to a parallel execution code; and

second detecting means for detecting, when a real address has been assigned to a parallel execution code, an instruction including the real address that is not capable of being expressed by an original word length of the instruction,

the flag setting means setting the boundary flag at a unit field located one of before and after unit fields to which the instruction detected by the second detecting means has been assigned.

6. The instruction conversion apparatus of claim **5**, further comprising:

replacing means for replacing an instruction detected by the second detecting means with a transfer instruction that transfers an address to a register and an addressing instruction that performs the same processing as the replaced instruction using the register,

the assigning means assigning the two instructions substituted by the replacing means to a plurality of unit fields, and

the flag setting means setting a boundary flag at one of the plurality of unit fields to which the two substituted instructions have been assigned to show a parallel execution boundary.

7. A processor, comprising:

fetch means for successively fetching parallel execution codes that include a plurality of unit fields from outside the processor;

a register set for storing a combination of a plurality of instructions included in at least two parallel execution codes that have been fetched by the fetch means;

decoding means for decoding, when the combination of instructions stored in the register set satisfies predetermined restrictions, the instructions in the combination in parallel; and

operation execution means for executing a plurality of operations in parallel in accordance with a decoding result of the decoding means;

s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch means,

the decoding means including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers and decode at least one opcode stored in any of the 1^{st} to s^{th} registers, and

the operation executing means being connected to the s+k−1 registers and executing operations in accordance with a decoding result of the s decoders.

8. The processor of claim **7**,

wherein a long instruction whose word length is equal to at least two but no more than k unit fields is stored in any of the s+k−1 registers with a first of the at least two but no more than k unit fields storing an opcode of the long instruction,

the decoding means including:

a decoding control unit which, when an opcode of a long instruction in stored in a u^{th} ($1<u<s$) unit field between the 1^{st} unit field the s^{th} unit field, has the u^{th} decoder decode the opcode stored in the u^{th} register and a value stored between the u^{th} register and the $(u+k-1)^{th}$ register outputted to the operation execution means as an operand of the long instruction.

9. The processor of claim **7**,

wherein the first unit field that stores the opcode of the long instruction has a format flag set at ON to show that the unit field forms part of a long instruction,

the decoding control unit detecting a register that stores a unit field whose format flag is set at ON as the u^{th} register, and

the decoding control unit having the uth decoder decode an operand stored in the u^{th} register and having a stored value between the u^{th} register and the $(u+k-1)^{th}$ register outputted to the operation execution means as an operand of the long instruction.

10. The processor of claim **7**,

wherein the decoding control unit performs control to invalidate a decoding operation of every decoder from the $(u+1)^{th}$ decoder onwards when a value stored between the $(u+1)^{th}$ register and the $(u+k-1)^{th}$ register is outputted to the operation execution means as an operand of a long instruction.

11. A recording medium storing executable code for a processor, the processor including (1) a fetch means for successively fetching parallel execution codes that each include a plurality of unit fields from outside the target processor, (2) s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch means, (3) decoding means, including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers, the decoders decoding at least one opcode stored in any of the 1^{st} to s^{th} registers, and (4) operation executing means, connected to the s+k−1 registers for executing operations in accordance with a decoding result of the s decoders,

the executable code stored on the recording medium being arranged such that at least one of an opcode and an operand of a long instruction having a word length of at least two but no more than k unit fields is arranged into to a u^{th} (where u is any integer such that $1 \leqq u \leqq s$) unit field between the 1^{st} unit field and the s^{th} unit field, and only an operand of the long instruction is arranged in unit fields from a $(u+1)^{th}$ unit field to a $(u+k-1)^{th}$ unit field.

12. A computer-readable recording medium storing an instruction conversion program that converts an instruction sequence into parallel execution codes that are executable by a target processor, the target processor having predetermined limitations regarding combinations of instructions that can be executed in parallel,

the instruction conversion program comprising:

an assigning step for successively assigning instructions in the instruction sequence to parallel execution codes; and

a control step for controlling the assigning step so that a combination of a plurality of instructions that have already been assigned to a parallel execution code and an instruction that the assigning step is about to assign to the parallel execution code satisfy the predetermined limitations of the target processor;

wherein the target processor includes (1) a fetch means for successively fetching parallel execution codes that each include a plurality of unit fields from outside the target processor, (2) s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch means, (3) decoding means, including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers, the decoders decoding at least one opcode

stored in any of the 1^{st} to s^{th} registers, and (4) operation executing means, connected to the s+k−1 registers for executing operations in accordance with a decoding result of the s decoders,

the assigning step assigning, when instructions to be assigned to a parallel execution code include a long instruction whose word length is equal to at least two but no more than k unit fields, at least one of an opcode and an operand of the long instruction to a u^{th} (where u is any integer such that 1<u<s) unit field between the 1^{st} unit field the s^{th} unit field, and only an operand of the long instruction to unit fields from a $(u+1)^{th}$ unit field to a $(u+k-1)^{th}$ unit field.

13. The computer-readable recording medium of claim 12,

wherein the instruction conversion program further comprises:

a grouping step for forming an instruction group of a plurality of instructions that do not exhibit a dependency relation (hereafter "data dependency relation"), a data dependency relation being a relation between an instruction defining a resource and an instruction referring to the same resource; and

a first detecting step for detecting, when a 1^{st} to an s^{th} unit field in a parallel execution code have been assigned at least one instruction by the assigning step and an instruction (hereafter "short instruction") with a shorter word length than a long instruction is left in the instruction group, a long instruction assigned to unit fields between the 1^{st} unit field and the s^{th} unit field,

wherein the control step includes a first control substep for controlling the assigning step to rearrange instructions that have already been assigned to the parallel execution code so that the detected long instruction is assigned to unit fields between the s^{th} unit field and the $(s+k-1)^{th}$ unit field and the short instruction remaining in the instruction group is assigned to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

14. The computer-readable recording medium of claim 13,

wherein the instruction group includes instructions that exhibit an anti-dependence and instructions that exhibit an output dependence, an anti-dependence being a relation between an instruction that refers to a resource and an instruction that thereafter defines the resource, and an output dependence being a relation between an instruction that defines a resource and another instruction that defines the resource,

the control step including a search substep for searching for a combination pattern, composed of a plurality of instructions in the instruction group, that is unaffected by an anti-dependence and an output dependence, and

the first control substep controlling the assigning step to rearrange the plurality of instructions in accordance with the combination pattern found by the search substep, to assign the long instruction found by the detecting step to unit fields from the s^{th} unit field to the $(s+k-1)^{th}$ unit field, and to assign a short instruction left in the instruction group to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

15. The computer-readable recording medium of claim 14,

wherein the instruction conversion program further comprises:

a flag setting step for setting a parallel execution boundary flag at each boundary that marks a position at which the predetermined limitations of the target processor dictate that parallel execution is not possible.

16. The computer-readable recording medium of claim 15,

wherein the instruction conversion program further comprises:

an address resolving step for assigning a real address to a parallel execution code; and

a second detecting step for detecting, when a real address has been assigned to a parallel execution code, an instruction including the real address that cannot be expressed by an original word length of the instruction,

the flag setting step setting the boundary flag at a unit field located one of before and after unit fields to which the instruction detected by the second detecting step has been assigned.

17. The computer-readable recording medium of claim 16,

wherein the instruction conversion program further comprises:

a replacing step for replacing an instruction detected by the second detecting step with a transfer instruction that transfers an address to a register and an addressing instruction that performs the same processing as the replaced instruction using the register,

the assigning step assigning the two instructions substituted by the replacing step to a plurality of unit fields, and

the flag setting step setting a boundary flag at one of the plurality of unit fields to which the two substituted instructions have been assigned to show a parallel execution boundary.

18. An instruction conversion apparatus that converts an instruction sequence into parallel execution codes that are executable by a target processor, the target processor having predetermined limitations regarding combinations of instructions capable of being executed in parallel,

the instruction conversion apparatus comprising:

an assigning unit for successively assigning instructions in the instruction sequence to parallel execution codes; and

a control unit for controlling the assigning unit so that a combination of a plurality of instructions that have already been assigned to a parallel execution code and an instruction that the assigning unit is about to assign to the parallel execution code satisfy the predetermined limitations of the target processor;

wherein the target processor includes (1) a fetch unit for successively fetching parallel execution codes that each include a plurality of unit fields from outside the target processor, (2) s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch unit, (3) a decoding unit, including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers, the decoders decoding at least one opcode stored in any of the 1^{st} to s^{th} registers, and (4) an operation executing unit, connected to the s+k−1 registers for executing operations in accordance with a decoding result of the s decoders,

the assigning unit assigning, when instructions to be assigned to a parallel execution code include a long

instruction whose word length is equal to at least two but no more than k unit fields, one of an opcode and an operand of the long instruction to a u^{th} (where u is any integer such that 1<u<s) unit field between the 1^{st} unit field and the s^{th} unit field, and only an operand of the long instruction to unit fields from a $(u+l)^{th}$ unit field to a $(u+k-1)^{th}$ unit field.

19. The instruction conversion apparatus of claim **18**, further comprising:

a grouping unit for forming an instruction group of a plurality of instructions that do not exhibit a dependency relation (hereafter "data dependency relation"), a data dependency relation being a relation between an instruction defining a resource and an instruction referring to the same resource; and

a first detecting unit for detecting, when a 1^{st} to an s^{th} unit field in a parallel execution code have been assigned at least one instruction by the assigning unit and an instruction (hereafter "short instruction") with a shorter word length than a long instruction is left in the instruction group, a long instruction assigned to unit fields between the 1^{st} unit field and the s^{th} unit field,

wherein the control unit includes a first control unit for controlling the assigning unit to rearrange instructions that have already been assigned to the parallel execution code so that the detected long instruction is assigned to unit fields between the s^{th} unit field and the $(s+k-l)^{th}$ unit field and the short instruction remaining in the instruction group is assigned to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

20. The instruction conversion apparatus of claim **19**,

wherein the instruction group includes instructions that exhibit an anti-dependence and instructions that exhibit an output dependence, an anti-dependence being a relation between an instruction that refers to a resource and an instruction that thereafter defines the resource, and an output dependence being a relation between an instruction that defines a resource and another instruction that defines the resource,

the control unit including a search unit for searching for a combination pattern, composed of a plurality of instructions in the instruction group, that is unaffected by an anti-dependence and an output dependence, and

the first control unit controlling the assigning unit to rearrange the plurality of instructions in accordance with the combination pattern found by the search unit, to assign the long instruction found by the detecting unit to unit fields from the s^{th} unit field to the $(s+k-1)^{th}$ unit field, and to assign a short instruction left in the instruction group to a unit field between the 1^{st} unit field and the $(s-l)^{th}$ unit field.

21. The instruction conversion apparatus of claim **20**, further comprising:

a flag setting unit for setting a parallel execution boundary flag at each boundary that marks a position at which the predetermined limitations of the target processor dictate that parallel execution is not possible.

22. The instruction conversion apparatus of claim **21**, further comprising:

an address resolving unit for assigning a real address to a parallel execution code; and

a second detecting unit for detecting, when a real address has been assigned to a parallel execution code, an instruction including the real address that is not capable of being expressed by an original word length of the instruction,

the flag setting unit setting the boundary flag at a unit field located one of before and after unit fields to which the instruction detected by the second detecting unit has been assigned.

23. The instruction conversion apparatus of claim **22**, further comprising:

a replacing unit for replacing an instruction detected by the second detecting unit with a transfer instruction that transfers an address to a register and an addressing instruction that performs the same processing as the replaced instruction using the register,

the assigning unit assigning the two instructions substituted by the replacing unit to a plurality of unit fields, and

the flag setting unit setting a boundary flag at one of the plurality of unit fields to which the two substituted instructions have been assigned to show a parallel execution boundary.

24. A processor, comprising:

a fetch unit for successively fetching parallel execution codes that include a plurality of unit fields from outside the processor;

a register set for storing a combination of a plurality of instructions included in at least two parallel execution codes that have been fetched by the fetch unit;

a decoding unit for decoding, when the combination of instructions stored in the register set satisfies predetermined restrictions, the instructions in the combination in parallel; and

an operation execution unit for executing a plurality of operations in parallel in accordance with a decoding result of the decoding unit;

s+k-1 (where s,k are integers no smaller than 2) registers for storing s+k-1 unit fields included in at least two parallel execution codes that have been fetched by the fetch unit,

the decoding unit including s decoders that correspond to 1^{st} to s^{th} registers in the s+k-1 registers and decode at least one opcode stored in any of the 1^{st} to s^{th} registers, and

the operation executing unit being connected to the s+k-1 registers and executing operations in accordance with a decoding result of the s decoders.

25. The processor of claim **24**,

wherein a long instruction whose word length is equal to at least two but no more than k unit fields is stored in any of the s+k-1 registers with a first of the at least two but no more than k unit fields storing an opcode of the long instruction,

the decoding unit including:

a decoding control unit which, when an opcode of a long instruction in stored in a u^{th} (1<u<s) unit field between the 1^{st} unit field the s^{th} unit field, has the u^{th} decoder decode the opcode stored in the u^{th} register

and a value stored between the u^{th} register and the $(u+k-1)^{th}$ register outputted to the operation execution unit as an operand of the long instruction.

26. The processor of claim 24

wherein the decoding control unit performs control to invalidate a decoding operation of every decoder from the $(u+1)^{th}$ decoder onwards when a value stored between the $(u+1)^{th}$ register and the $(u+k-1)^{th}$ register is outputted to the operation execution unit as an operand of a long instruction.

27. A recording medium storing executable code for a processor, the processor including (1) a fetch unit for successively fetching parallel execution codes that each include a plurality of unit fields from outside the target processor, (2) s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch unit, (3) a decoding unit, including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers, the decoders decoding at least one opcode stored in any of the 1^{st} to s^{th} registers, and (4) an operation executing unit, connected to the s+k−1 registers for executing operations in accordance with a decoding result of the s decoders,

the executable code stored on the recording medium being arranged such that at least one of an opcode and an operand of a long instruction having a word length of at least two but no more than k unit fields is arranged into to a u^{th} (where u is any integer such that $1<u<s$) unit field between the 1^{st} unit field and the s^{th} unit field and the s^{th} unit field, and only an operand of the long instruction is arranged in unit fields from a $(u+1)^{th}$ unit field to a $(u+k-1)^{th}$ unit field.

28. A computer-readable recording medium storing an instruction conversion program that converts an instruction sequence into parallel execution codes that are executable by a target processor, the target processor having predetermined limitations regarding combinations of instructions that can be executed in parallel,

the instruction conversion program comprising:

an assigning step for successively assigning instructions in the instruction sequence to parallel execution codes; and

a control step for controlling the assigning step so that a combination of a plurality of instructions that have already been assigned to a parallel execution code and an instruction that the assigning step is about to assign to the parallel execution code satisfy the predetermined limitations of the target processor;

wherein the target processor includes (1) a fetch unit for successively fetching parallel execution codes that each include a plurality of unit fields from outside the target processor, (2) s+k−1 (where s,k are integers no smaller than 2) registers for storing s+k−1 unit fields included in at least two parallel execution codes that have been fetched by the fetch unit, (3) a decoding unit, including s decoders that correspond to 1^{st} to s^{th} registers in the s+k−1 registers, the decoders decoding at least one opcode stored in any of the 1^{st} to s^{th} registers, and (4) an operation executing unit, connected to the s+k−1 registers for executing operations in accordance with a decoding result of the s decoders,

the assigning step assigning, when instructions to be assigned to a parallel execution code include a long instruction whose word length is equal to at least two but no more than k unit fields, at least one of an opcode and an operand of the long instruction to a u^{th} (where u is any integer such that $1<u<s$) unit field between the 1^{st} unit field the s^{th} unit field, and only an operand of the long instruction to unit fields from a $(u+1)^{th}$ unit field to a $(u+k-1)^{th}$ unit field.

29. The computer-readable recording medium of claim 28,

wherein the instruction conversion program further comprises:

a grouping step for forming an instruction group of a plurality of instructions that do not exhibit a dependency relation (hereafter "data dependency relation"), a data dependency relation being a relation between an instruction defining a resource and an instruction referring to the same resource; and

a first detecting step for detecting, when a 1^{st} to an s^{th} unit field in a parallel execution code have been assigned at least one instruction by the assigning step and an instruction (hereafter "short instruction") with a shorter word length than a long instruction is left in the instruction group, a long instruction assigned to unit fields between the 1^{st} unit field and the s^{th} unit field,

wherein the control step includes a first control substep for controlling the assigning step to rearrange instructions that have already been assigned to the parallel execution code so that the detected long instruction is assigned to unit fields between the s^{th} unit field and the $(s+k-1)^{th}$ unit field and the short instruction remaining in the instruction group is assigned to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

30. The computer-readable recording medium of claim 29,

wherein the instruction group includes instructions that exhibit an anti-dependence and instructions that exhibit an output dependence, an anti-dependence being a relation between an instruction that refers to a resource and an instruction that thereafter defines the resource, and an output dependence being a relation between an instruction that defines a resource and another instruction that defines the resource,

the control step including a search substep for searching for a combination pattern, composed of a plurality of instructions in the instruction group, that is unaffected by an anti-dependence and an output dependence, and

the first control substep controlling the assigning step to rearrange the plurality of instructions in accordance with the combination pattern found by the search substep, to assign the long instruction found by the detecting step to unit fields from the s^{th} unit field to the $(s+k-1)^{th}$ unit field, and to assign a short instruction left in the instruction group to a unit field between the 1^{st} unit field and the $(s-1)^{th}$ unit field.

31. The computer-readable recording medium of claim 30,

wherein the instruction conversion program further comprises:

a flag setting step for setting a parallel execution boundary flag at each boundary that marks a position at which the predetermined limitations of the target processor dictate that parallel execution is not possible.

32. The computer-readable recording medium of claim 31,

wherein the instruction conversion program further comprises:

an address resolving step for assigning a real address to a parallel execution code; and

a second detecting step for detecting, when a real address has been assigned to a parallel execution code, an instruction including the real address that cannot be expressed by an original word length of the instruction,

the flag setting step setting the boundary flag at a unit field located one of before and after unit fields to which the instruction detected by the second detecting step has been assigned.

33. The computer-readable recording medium of claim 32,

wherein the instruction conversion program further comprises:

a replacing step for replacing an instruction detected by the second detecting step with a transfer instruction that transfers an address to a register and an addressing instruction that performs the same processing as the replaced instruction using the register,

the assigning step assigning the two instructions substituted by the replacing step to a plurality of unit fields, and

the flag setting step setting a boundary flag at one of the plurality of unit fields to which the two substituted instructions have been assigned to show a parallel execution boundary.

* * * * *